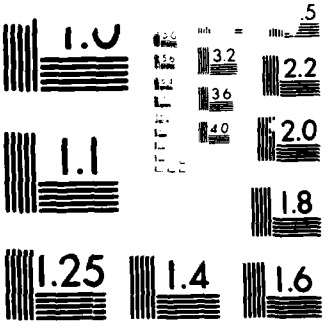
AD-A168 698 A NOTE ON NONLINEAR APPROACHES TO LINEAR PROGRAMMING 1/1
(U) STANFORD UNIV CA SYSTEMS OPTIMIZATION LAB
P E GILL ET AL. APR 86 SOL-86-7 N00014-85-K-0343
UNCLASSIFIED F/G 12/1 NL



MICROCOPY RESOLUTION TEST

(12)

Systems
Optimization
Laboratory

A NOTE ON NONLINEAR APPROACHES TO LINEAR PROGRAMMING

by

Philip E. Gill, Walter Murray,
Michael A. Saunders and Margaret H. Wright

TECHNICAL REPORT SOL 86-7

April 1986

Department of Operations Research
Stanford University
Stanford, CA 94305

12

SYSTEMS OPTIMIZATION LABORATORY
DEPARTMENT OF OPERATIONS RESEARCH
STANFORD UNIVERSITY
STANFORD, CALIFORNIA 94305-4022

A NOTE ON NONLINEAR APPROACHES TO LINEAR PROGRAMMING

by

Philip E. Gill, Walter Murray,
Michael A. Saunders and Margaret H. Wright

TECHNICAL REPORT SOL 86-7

April 1986

Research and reproduction of this report were partially supported by the U.S. Department of Energy Contract DE-AA03-76SF00326, PA# DE-AS03-76ER72018; and Office of Naval Research Contract N00014-85-K-0343.

A NOTE ON
NONLINEAR APPROACHES TO LINEAR PROGRAMMING†

Philip E. Gill, Walter Murray,
Michael A. Saunders and Margaret H. Wright

Systems Optimization Laboratory
Department of Operations Research
Stanford University
Stanford, California 94305

Technical Report SOL 86-7
April 1986

ABSTRACT

Certain new approaches to linear programming have recently received considerable publicity because of the promise of substantial improvements in efficiency compared to the simplex method. This note briefly discusses several research directions in methods for solving linear programs using nonlinear problem transformations. In particular, we describe application of a barrier transformation to the dual, and the development of sparse least-squares methods based on the *LU* factorization of the least-squares matrix or its transpose.

The material contained in this report is based upon research supported by the U.S. Department of Energy Contract DE-AA03-76SF00326, PA No. DE-AS03-76ER72018; and the Office of Naval Research Contract N00014-85-K-0343.

† Presented at the Workshop on Iterative Methods for Mathematical Programs, Stanford University, April 17 18, 1986.

1. Introduction

Linear programming (LP) has been the subject of great interest recently because of publicity surrounding an algorithm (Karmarkar, 1984) that is not only polynomial in complexity, but also is claimed to be *much faster* than the simplex method, developed nearly 40 years ago by George B. Dantzig. Since the publication of Karmarkar's original algorithm, interest has continued to grow in alternatives to the simplex method (see, e.g., Gay, 1985; Goldfarb and Mehrotra, 1985; Osborne, 1986; Shanno and Marsten, 1985; Todd and Burrell, 1985; Tomlin, 1985; Vanderbei, Meketon and Freedman, 1985). In this connection, Gill *et al.* (1985) have shown that Karmarkar's projective method is closely related to the classical logarithmic barrier function. In particular, Karmarkar's projective method is a special case of a projected Newton method applied to the logarithmic barrier function. Based on this equivalence, many researchers are re-examining the effectiveness of once-discarded methods that in effect "nonlinearize" a linear program. The proof of polynomial complexity of Karmarkar's original algorithm suggests that a suitable nonlinear transformation may somehow overcome the inherently combinatorial nature of the simplex method.

Because the main interest in nonlinear LP techniques arises from their potential speed, an essential part of the work reported in Gill *et al.* (1985) involved numerical experiments with an initial implementation of the barrier method on a set of small- to medium-scale test problems. The results were encouraging in that the barrier method was comparable in speed to the simplex method on certain problems. Typically, a barrier algorithm converges in a smaller number of iterations than the simplex method. However, since each iteration of the barrier method requires solution of a linear least-squares subproblem, a barrier method will be faster than the simplex method only if these subproblems can be solved quickly. Thus, the hope is that barrier-type methods will be faster than the simplex method—even substantially so—for problems with structure that allows fast solution of the least-squares subproblems.

The remainder of this note is organized as follows. In Section 2, we summarize some background on the application of a logarithmic barrier function method to a linear program in standard form. A disadvantage of the primal formulation is that the search direction is restricted to lie in a subspace. In order to satisfy this requirement, either the associated linear least-squares subproblem must be solved accurately, or the vector must be projected into the appropriate subspace. As an alternative, Section 3 gives the dual formulation of an LP, which leads to a purely unconstrained subproblem, thereby allowing approximate methods to be used to solve the least-squares subproblem. In Section 4, we discuss techniques for solving the least-squares subproblem, and consider possible uses of the LU factorization.

2. A Barrier-Function Method for the Primal LP

This section is intended to serve as general background on barrier functions and on the work described in Gill *et al.* (1985).

Barrier-function methods treat *inequality* constraints by creating a ***barrier function***, which is a combination of the original objective function and a weighted sum of functions with a positive

singularity at the constraint boundary. (Many barrier functions have been proposed; we consider only the logarithmic barrier function, first suggested by Frisch, 1955.) Barrier-function methods require a strictly feasible starting point for each minimization, and generate a sequence of strictly feasible iterates. (For a complete discussion of barrier methods, see Fiacco, 1979; both barrier- and penalty-function methods are described in Fiacco and McCormick, 1968. Brief overviews are given in Fletcher, 1981, and Gill, Murray and Wright, 1981.)

In particular, consider the inequality-constrained problem

$$\begin{aligned} \underset{x \in \mathbb{R}^n}{\text{minimize}} \quad & \phi(x) \\ \text{subject to} \quad & h(x) \geq 0, \end{aligned} \tag{2.1}$$

where h is an ℓ-vector of nonlinear functions. Let x^* denote the solution of (2.1). With a barrier-function approach, the inequality constraints of (2.1) are "removed" by including transformed versions in a modified objective function, as follows. Given a *barrier parameter* μ ($\mu > 0$), let $x(\mu)$ denote the *unconstrained* minimum of

$$B(x, \mu) \equiv \phi(x) - \mu \sum_{i=1}^{\ell} \ln\bigl(h_i(x)\bigr).$$

Under mild conditions, it can be shown that

$$\lim_{\mu \to 0} x(\mu) = x^*.$$

Many results have been proved concerning the asymptotic properties of the sequence $\{x(\mu)\}$ as $\mu \to 0$ (see, e.g., Mifflin, 1972a, b; Jittorntrum, 1978; Jittorntrum and Osborne, 1978).

Now consider applying a barrier-function method to the following linear program in standard form (which will also be called the *primal* LP):

$$\underset{x \in \mathbb{R}^n}{\text{minimize}} \quad c^T x \tag{2.2a}$$

$$\text{subject to} \quad Ax = b \tag{2.2b}$$

$$x \geq 0, \tag{2.2c}$$

where A is an $m \times n$ matrix with $m \leq n$.

Since the barrier transformation may be applied only to the *inequality* constraints (2.2c), the subproblem associated with the primal LP treats the linear equality constraints directly and transforms only the bounds:

$$\begin{aligned} \underset{x \in \mathbb{R}^n}{\text{minimize}} \quad & F(x) \equiv c^T x - \mu \sum_{j=1}^{n} \ln x_j \\ \text{subject to} \quad & Ax = b. \end{aligned} \tag{2.3}$$

A standard approach to solving a linearly constrained problem of the form (2.3) is to use a *feasible-point descent method* (see, e.g., Gill, Murray and Wright, 1981). The current iterate x always satisfies $Ax = b$, and the next iterate $\bar{x}$ is defined as

$$\bar{x} = x + \alpha p, \tag{2.4}$$

where p is an n-vector (the *search direction*), and α is a positive scalar (the *steplength*). The computation of p and α must ensure that $A\bar{x} = b$ and $F(\bar{x}) < F(x)$.

The *Newton search direction* is defined as the step to the minimum of the quadratic approximation to $F(x)$ derived from the local Taylor series, subject to retaining feasibility. Thus, the Newton search direction is the solution of the following quadratic program:

$$\underset{p \in \mathbb{R}^n}{\text{minimize}} \quad g^T p + \tfrac{1}{2} p^T H p \tag{2.5a}$$

$$\text{subject to} \quad Ap = 0, \tag{2.5b}$$

where $g \equiv \nabla F(x)$ and $H \equiv \nabla^2 F(x)$. The optimality conditions for (2.5) imply that

$$g + Hp = A^T \pi \tag{2.6}$$

for a vector π of Lagrange multipliers. Thus, p and π satisfy the partitioned linear system:

$$\begin{pmatrix} H & A^T \\ A & 0 \end{pmatrix} \begin{pmatrix} -p \\ \pi \end{pmatrix} = \begin{pmatrix} g \\ 0 \end{pmatrix}. \tag{2.7}$$

If H is nonsingular, one method of solving (2.7) is to solve the equations

$$AH^{-1}A^T \pi = AH^{-1} g, \tag{2.8a}$$

$$Hp = A^T \pi - g. \tag{2.8b}$$

Let p_P (the Newton direction for the *primal* LP) denote the solution of subproblem (2.5) when F is the barrier function in (2.3), with associated Lagrange multiplier π_P. The derivatives of F are:

$$g(x) = c - \mu X^{-1} e \qquad \text{and} \qquad H(x) = \mu X^{-2}, \tag{2.9a}$$

where

$$X = \text{diag}\,(x_j), \quad j = 1, \ldots, n, \tag{2.9b}$$

and $e = (1, 1, \ldots, 1)^T$. Since $H(x)$ is positive definite when $x > 0$, p_P is finite and unique, and is a *descent direction* for $F(x)$, i.e., $(c - \mu X^{-1} e)^T p_P < 0$. Substituting from (2.9a) in (2.8a), we see that π_P satisfies

$$AX^2 A^T \pi_P = AX(Xc - \mu e). \tag{2.10}$$

Recall from standard linear algebra (e.g., Stewart, 1973) that, for any matrix C, the solution of the least-squares problem

$$\underset{x}{\text{minimize}} \ \|f - Cx\|_2^2 \tag{2.11}$$

satisfies the *normal equations*

$$C^T C x = C^T f, \tag{2.12}$$

which are always compatible. In referring to (2.11) and (2.12), we shall call C the *least-squares matrix* and $C^T C$ the *normal-equation matrix*.

Comparing (2.10) and (2.12) and taking $C = XA^T$ in (2.12), it follows that π_P solves the least-squares problem

$$\underset{\pi}{\text{minimize}} \ \|Xc - \mu e - XA^T\pi\|_2^2. \tag{2.13}$$

The vector p_P is then defined by

$$p_P = -(1/\mu)Xr_P, \tag{2.14}$$

where r_P is the optimal residual of (2.13):

$$r_P = Xc - \mu e - XA^T\pi_P. \tag{2.15}$$

Note that if m variables are nonzero at the solution of (2.2) (i.e., if the primal is nondegenerate), then XA^T will retain full rank even as the iterates approach the solution. (This contrasts with the structural ill-conditioning of the Hessian matrices of the barrier function when fewer than n constraints are active at the solution; see, e.g., Murray, 1971.) However, a well known feature of practical linear programs is that they usually are degenerate; i.e., fewer than m variables are nonzero at the solution. In this case, the matrix XA^T is rank-deficient at the solution, and becomes increasingly ill-conditioned as $\mu \to 0$. Therefore, any method used in practice to solve (2.13) must be able to cope with singularity and extreme ill-conditioning. (Numerical solution of the least-squares subproblems will be discussed in Section 4.)

3. Solving the Dual Problem

With the approach described in Section 2, the search direction is required to lie in a particular subspace in order for the iterates to retain feasibility with respect to the *equality* constraints (2.2*b*). In this section, we describe an alternative based on the *dual* linear program associated with (2.2):

$$\begin{aligned} &\underset{y \in \mathbb{R}^m}{\text{minimize}} && b^T y \\ &\text{subject to} && A^T y \geq -c. \end{aligned} \tag{3.1}$$

The optimal vector y^* for (3.1) is the negative of the Lagrange multiplier vector π^* for the equality constraints of (2.2). (Use of the dual LP in this context has been suggested independently by other researchers, for example at the recent ONR-sponsored Workshop on New Directions in Mathematical Programming, Monterey, California, February 20–21, 1986. It is also closely related to the work of Eriksson, 1981, 1985.)

Since all the constraints of (3.1) are inequalities, applying the barrier transformation to this problem gives a purely unconstrained subproblem:

$$\underset{y}{\text{minimize}} \ F(y) = b^T y - \mu \sum_{j=1}^{n} \ln(a_j^T y + c_j). \tag{3.2}$$

Let $y(\mu)$ denote the unconstrained minimum of (3.2). Defining

$$D = \text{diag}(d_j) \qquad \text{and} \qquad X = \text{diag}(x_j) = \mu D^{-1},$$

where $d_j = a_j^T y + c_j$ and $x_j = \mu/d_j$, we have

$$\nabla F = b - Ax \qquad \text{and} \qquad \nabla^2 F = \frac{1}{\mu} AX^2 A^T.$$

The gradient of F must vanish at $y(\mu)$, which implies that

$$\sum_{j=1}^{n} \frac{\mu}{a_j^T y(\mu) + c_j} a_j = b, \tag{3.3a}$$

and hence

$$\lim_{\mu \to 0} \frac{\mu}{d_j} = x_j^* \ . \tag{3.3b}$$

In order to define a Newton method for solving (3.2), we assume that a point y is available for which $A^T y + c > 0$ (so that $\nabla^2 F$ is positive definite). Using the above derivatives, p_D (the Newton search direction for the *dual*) then satisfies the system

$$AX^2 A^T p_D = \mu(Ax - b). \tag{3.4}$$

In contrast to the subproblem (2.5) associated with the primal problem, p_D need not lie in any particular subspace, and hence (3.4) may be solved only *approximately*—for example, using a few iterations of a conjugate-gradient method.

Note that if the right-hand side of (3.4) can be expressed in the form AXv for some vector v, (3.4) will have the form (2.12) of the normal equations for a least-squares problem. (It is advantageous numerically to solve a least-squares problem rather than equations of the form (3.4). However, note that (3.4) always has a bounded solution if a primal-feasible point exists.) We therefore seek v such that

$$AXv = Ax - b, \tag{3.5}$$

whereupon p_D becomes the solution of the least-squares problem

$$\underset{p}{\text{minimize}} \ \|\mu v - XA^T p\|_2^2, \tag{3.6}$$

which may be solved using the techniques to be described in Section 4. Note that $x > 0$, whereas $A(x - Xv) = b$. As convergence occurs, $v \to 0$ and $x \to x^*$.

It might appear that the elements of the least-squares matrix would become unbounded with this approach, in contrast to the primal. However, the relationship (3.3) shows that $d_j \to 0$ only as $\mu \to 0$, and thus the limiting matrices X and XA^T should remain bounded.

An alternative to the exact dual formulation (3.4) is to retain the same "Hessian" for several iterations. In Newton-type methods for nonlinear optimization, it is common to define the search direction with a positive-definite *approximation* to the Hessian matrix. (For example, the exact Hessian may be indefinite or expensive to compute.) In the present context, any positive-definite

X may be used in (3.4) (3.6) to generate a descent direction. It is essential to retain the definition $x = \mu D^{-1}e$ to obtain the correct gradient $b - Ax$, but X need not be defined as μD^{-1}.

An immediate implication is that same matrix X could be used for perhaps several iterations of the Newton-type method. (Convergence results for methods for this type are given in, e.g., Dennis, 1970, and Ortega and Rheinboldt, 1970.) Since a sparse factorization of XA^T is needed to compute v and p in (3.5)-(3.6), considerable economies can be made by retaining the factorization from the previous iteration.

4. Solving the Least-Squares Subproblem

It should be emphasized that the least-squares subproblems associated with the primal and dual linear programs ((2.13) and (3.6)) are very similar, and it is crucial that they be solved efficiently. In this section we review techniques that have been suggested or used, and then describe some new approaches based on the LU factorization.

4.1. Background

The major strategies for solving sparse least-squares problems include:

- *direct* methods, based on a sparse QR factorization of C (e.g., George and Heath, 1980) or on the Cholesky factorization of the normal-equation matrix C^TC (e.g., George and Liu, 1981);
- *updating* methods that factorize *most* of C or C^TC and deal with a few omitted rows or columns by partitioning (e.g., Heath, 1984);
- purely *iterative* methods that perform no factorization, such as a straightforward conjugate-gradient method;
- *hybrid* methods that combine factorization and iterative techniques. For example, in Gill *et al.* (1985), the subproblem (2.13) is solved with a preconditioned version of the stabilized conjugate-gradient method LSQR (Paige and Saunders, 1982), where the preconditioner is the Cholesky factor of a sparse matrix related to AX^2A^T. (The use of a preconditioned conjugate-gradient method in this context is also discussed by Gay, 1985.) For early work on preconditioners, see Axelsson (1974) and Concus, Golub and O'Leary (1976). Some techniques for computing an approximate ("incomplete") Cholesky factor based on sparsity considerations are given in, e.g., Meijerink and van der Vorst (1977), Manteuffel (1980) and Munksgaard (1980).

The effectiveness of these options varies with the problem.

When using an iterative scheme, it is desirable to be able to use only an *approximate* solution (obtained, say, from a truncated conjugate-gradient method; see, e.g., Dembo, Eisenstat and Steihaug, 1982). In this context, a disadvantage of the primal formulation is that an approximate solution of (2.13) will not satisfy the equality constraints (2.5*b*). One means of overcoming this difficulty is to *project* the search direction into the null space of A, for example using a sparse

representation of a basis for the null space. Such a matrix may be constructed from an LU factorization of A (which would need to be computed only once), or alternatively it may be obtained from an LU factorization of XA^T.

For the dual formulation, a truncated conjugate-gradient method is most promising because an approximate solution of (3.4) can be shown to be a descent direction, as required. However, the speed of convergence of the conjugate-gradient method is crucial to practical success. We now consider methods for preconditioning the conjugate-gradient method using an LU factorization.

4.2. The LU factorization

Given a general sparse matrix C with n rows and m columns, we write

$$C = LU, \tag{4.1}$$

where L is square and nonsingular, the dimensions of U are those of C, and L and U are (nominally) unit-lower-triangular and upper-triangular respectively. (In general, the rows and columns of C will have been permuted during the factorization, but we shall ignore the permutations for simplicity of notation.) If a suitable form of threshold pivoting (see Section 4.4) is used in computing L and U, L tends to remain well-conditioned throughout, so that the condition of C is almost always reflected in U.

A set of procedures (LUSOL; see Gill *et al.*, 1986) has recently been developed for computing and updating the LU factors of a general sparse matrix; the techniques in LUSOL are related to the work of Reid (1976, 1982) on computing a sparse LU factorization and performing sparse Bartels-Golub updates (Bartels and Golub, 1969). LUSOL provides a means for implementing several methods to be described below.

In the problems of interest, C is of the form XA^T and $n > m$. Hence (4.1) has the form

$$C = XA^T = LU = \hat{L}\hat{U}, \qquad \hat{L} = L\begin{pmatrix} I \\ 0 \end{pmatrix}, \qquad U = \begin{pmatrix} \hat{U} \\ 0 \end{pmatrix}, \tag{4.2}$$

where $\hat{L}$ denotes the first m columns of L and $\hat{U}$ is an $m \times m$ upper-triangular matrix. (The last $(n - m)$ columns of L will simply be those of the identity.) Again $\hat{L}$ should be well-conditioned, while $\hat{U}$ reflects the condition of C. For the present we assume that C has full rank, so that $\hat{U}$ is nonsingular.

Note that (4.2) can be used to find the vector v in (3.5) needed to formulate the dual as the least-squares subproblem (3.6). It is sufficient to solve the two systems

$$\hat{U}^T u = A\hat{x} - b \qquad \text{and} \qquad L^T v = \begin{pmatrix} u \\ 0 \end{pmatrix}.$$

4.3. LU preconditioning

Following Peters and Wilkinson (1970) and Björck (1976), we note that the least-squares problem

$$\underset{p}{\text{minimize}} \ \|f - Cp\|_2^2 \tag{4.3}$$

may be solved using a factorization of the form (4.2), by solving

$$\underset{q}{\text{minimize}} \ \|f - \hat{L}q\|_2^2, \tag{4.4a}$$

$$\hat{U}p = q. \tag{4.4b}$$

Because $\hat{L}$ is well-conditioned, an iterative method may converge more rapidly on problem (4.4*a*) than on the original problem (4.3) (unless C itself is well-conditioned or has clustered singular values). Some experiments with this approach are described in Saunders (1979); for related work, see Delves and Barrodale (1979).

In the present context, greater efficiency is likely to result from further preconditioning, particularly as the solution is approached. If $\bar{L}$ is the first m rows of $\hat{L}$:

$$\hat{L} = \begin{pmatrix} \bar{L} \\ M \end{pmatrix},$$

it follows that p may be determined by solving

$$\underset{s}{\text{minimize}} \ \|f - \begin{pmatrix} I \\ M\bar{L}^{-1} \end{pmatrix} s\|_2^2, \tag{4.5a}$$

$$\bar{L}q = s, \qquad \hat{U}p = q. \tag{4.5b}$$

The nature of the linear programming problem and the pivoting strategy used in LUSOL are such that $M\bar{L}^{-1} \rightarrow 0$ as $y \rightarrow y^*$. *Hence, the number of iterations required by a conjugate-gradient algorithm should become small as the solution is approached.*

In fact, the factorization $XA^T = LU$ obtained with the LUSOL pivoting strategy serves to partition X and A in the form

$$XA^T = \begin{pmatrix} X_B & \\ & X_N \end{pmatrix} \begin{pmatrix} B^T \\ N^T \end{pmatrix} = \begin{pmatrix} X_B B^T \\ X_N N^T \end{pmatrix},$$

and (4.5) is equivalent to solving the original problem (4.3) with the preconditioner $X_B B^T = \bar{L}\hat{U}$. At all stages, the partitioned factorization pinpoints a nonsingular B that corresponds to the usual basis matrix in the simplex method, and ultimately it will correspond to an optimal basis.

In general, it may not be necessary to recompute the LU factorization at every iteration. If the current factors are $X_k A^T = L_k U_k$, the factors of $X_{k+1}A^T$ could be taken as $L_{k+1} = \tilde{D}L_k$ and $U_{k+1} = U_k$, where $\tilde{D} = X_{k+1}X_k^{-1}$, as long as $\tilde{D}$ is reasonably well-conditioned. In some cases it may be efficient to update the LU factors to account for those elements of $\tilde{D}$ that introduce ill-conditioning, using some of the update procedures available in LUSOL.

Another approach is to compute the LU factorization of C^T, so that

$$C^T = AX = LU, \tag{4.6}$$

where L is nonsingular and U is an $m \times n$ trapezoid. In this case, LUSOL will need to be enhanced to use a threshold form of *complete pivoting*, in order to ensure that the first m columns of U point to a suitable basis matrix B. If a matrix C has full column rank, a reliable factorization $C = LU$ can almost always be obtained using Gaussian elimination with *threshold partial pivoting*. By this we mean that when a potential pivot is selected from the remaining rows and columns, it must be reasonably large relative to other elements in the same column, but it need not be compared with elements in other columns. Virtually all existing LU software uses this strategy.

When computing the factorization (4.6), the column rank will be drastically deficient, and the triangular part of U will reflect the true rank only if pivots are chosen to be reasonably large relative to *all* remaining elements; i.e., only if *threshold complete pivoting* is implemented. Such a strategy requires keeping track of the largest remaining element at each stage without excessive overhead. Any such enhancement to LUSOL would carry a benefit whether C or C^T is being factorized, since XA^T is expected to be rank-deficient.

5. References

Axelsson, O. (1974). On preconditioning and convergence acceleration in sparse matrix problems, Report 74-10, CERN European Organization for Nuclear Research, Geneva.

Bartels, R. H. and Golub, G. H. (1969). The simplex method of linear programming using the *LU* decomposition, *Communications of the Association for Computing Machinery* 12, 266–268.

Björck, Å. (1976). Methods for sparse least squares problems, in J. R. Bunch and D. J. Rose (eds.), *Sparse Matrix Computations*, Academic Press, London and New York, 177–199.

Concus, P., Golub, G. H. and O'Leary, D. P. (1976). A generalized conjugate-gradient method for the numerical solution of elliptic partial differential equations, in J. R. Bunch and D. J. Rose (eds.), *Sparse Matrix Computations*, Academic Press, London and New York, 309–332.

Delves, L. M. and Barrodale, I. (1979). A fast direct method for the least squares solution of slightly overdetermined sets of linear equations, *J. Institute of Mathematics and its Applications* 24, 149-156.

Dembo, R. S., Eisenstat, S. C. and Steihaug T. (1982). Inexact Newton methods, *SIAM J. on Numerical Analysis* 19, 400–408.

Dennis, J. E., Jr. (1970). On the convergence of Newton-like methods, in P. Rabinowitz (ed.), *Numerical Methods for Nonlinear Algebraic Equations*, Gordon and Breach, London and New York, 163–181.

Eriksson, J. (1981). *Algorithms for entropy and mathematical programming*, Ph. D. Thesis, Department of Mathematics, Linköping University, Linköping, Sweden.

Eriksson, J. (1985). An iterative primal-dual algorithm for linear programming, Report LiTH-MAT-R-1985-10, Department of Mathematics, Linköping University, Sweden.

Fiacco, A. V. (1979). Barrier methods for nonlinear programming, in A. Holzman (ed.), *Operations Research Support Methodology*, Marcel Dekker, Inc., New York, 377 440.

Fiacco, A. V. and McCormick, G. P. (1968). *Nonlinear Programming: Sequential Unconstrained Minimization Techniques*, John Wiley and Sons, New York and Toronto.

Fletcher, R. (1981). *Practical Methods of Optimization, Volume 2*, John Wiley and Sons, Chichester and New York.

Frisch, K. R. (1955). The logarithmic potential method of convex programming, Memorandum of May 13, 1955, University Institute of Economics, Oslo, Norway.

Gay, D. M. (1985). A variant of Karmarkar's linear programming algorithm in standard form, *Numerical Analysis Manuscript* 85-10, AT&T Bell Laboratories.

George, J. A. and Heath, M. T. (1980). Solution of sparse linear least squares problems using Givens rotations, *Linear Algebra and its Applications* 34, 69–83.

George, J. A. and Liu, J. W. (1981). *Computer Solution of Large Sparse Positive Definite Systems*, Prentice-Hall, Englewood Cliffs, NJ.

Gill, P. E., Murray, W., Saunders, M. A. and Wright, M. H. (1984). Sparse matrix methods in optimization, *SIAM J. on Scientific and Statistical Computing* 5, 562–589.

Gill, P. E., Murray, W., Saunders, M. A., Tomlin, J. A. and Wright, M. H. (1985). On projected Newton barrier methods for linear programming and an equivalence to Karmarkar's projective method, Report SOL 85-11, Department of Operations Research, Stanford University.

Gill, P. E., Murray, W., Saunders, M. A. and Wright, M. H. (1986). Maintaining *LU* factors of a general sparse matrix, submitted to *Linear Algebra and its Applications.*

Gill, P. E., Murray, W. and Wright, M. H. (1981). *Practical Optimization*, Academic Press, London and New York.

Goldfarb, D. and Mehrotra, S. (1985). A relaxed version of Karmarkar's method, Technical Report, Department of Industrial Engineering and Operations Research, Columbia University, New York.

Heath, M. T. (1984). Numerical methods for large sparse linear least squares problems, *SIAM J. on Scientific and Statistical Computing* 5, 497–513.

Jittorntrum, K. (1978). *Sequential Algorithms in Nonlinear Programming*, Ph. D. Thesis, Australian National University.

Jittorntrum, K. and Osborne, M. R. (1978). Trajectory analysis and extrapolation in barrier function methods, *Journal of Australian Mathematical Society* 21, 1–18.

Karmarkar, N. (1984). A new polynomial-time algorithm for linear programming, *Combinatorica* 4, 373–395.

Manteuffel, T. A. (1980). An incomplete factorization technique for positive definite linear systems, *Mathematics of Computation* 34, 473-497.

Meijerink, J. A. and van der Vorst, H. A. (1977). An iterative solution method for linear systems in which the coefficient matrix is an M-matrix, *Mathematics of Computation* 31, 148-162.

Mifflin, R. (1972a). Convergence bounds for nonlinear programming algorithms, Administrative Sciences Report 57, Yale University, Connecticut.

Mifflin, R. (1972b). On the convergence of the logarithmic barrier function method, in F. A. Lootsma (ed.), *Numerical Methods for Non-Linear Optimization*, Academic Press, London and New York, 367-369.

Munksgaard, N. (1980). Solving sparse symmetric sets of linear equations by preconditioned conjugate gradients, *ACM Transactions on Mathematical Software* 6, 206–219.

Murray, W. (1971). Analytical expressions for the eigenvalues and eigenvectors of the Hessian matrices of barrier and penalty functions, *J. Optimization Theory and Applications* 7, 189–196.

Ortega, J. M. and Rheinboldt, W. C. (1970). *Iterative Solution of Nonlinear Equations in Several Variables*, Academic Press, London and New York.

Osborne, M. R. (1986). Dual barrier functions with superfast rates of convergence for the linear programming problem, Department of Statistics, Research School of Social Sciences, Australian National University.

Paige, C. C. and Saunders, M. A. (1982). *LSQR*: An algorithm for sparse linear equations and sparse least-squares, *ACM Transactions on Mathematical Software* 8, 43–71.

Peters, G. and Wilkinson, J. H. (1970). The least-squares problem and pseudo-inverses, *Computer Journal* 13, 309–316.

Reid, J. K. (1976). Fortran subroutines for handling sparse linear programming bases, Report AERE-R8269, Atomic Energy Research Establishment, Harwell, England.

Reid, J. K. (1982). A sparsity-exploiting variant of the Bartels-Golub decomposition for linear programming bases, *Mathematical Programming* 24, 55–69.

Saunders, M. A. (1979). Sparse least squares by conjugate gradients: a comparison of preconditioning methods, in J. F. Gentleman (ed.), *Computer Science and Statistics: 12th Annual Symposium on the Interface*, University of Waterloo, Waterloo, Ontario, Canada, 15–20.

Shanno, D. F. and Marsten, R. E. (1985). On implementing Karmarkar's method, Working Paper 85-1, Graduate School of Administration, University of California at Davis, California.

Stewart, G. W. (1973). *Introduction to Matrix Computations*, Academic Press, New York.

Todd, M. J. and Burrell, B. P. (1985). An extension of Karmarkar's algorithm for linear programming using dual variables, Report 648, School of Operations Research and Industrial Engineering, Cornell University, New York.

Tomlin, J. A. (1985). An experimental approach to Karmarkar's projective method for linear programming, Manuscript, Ketron Inc., Mountain View, California.

Vanderbei, R. J., Meketon, M. S. and Freedman, B. A. (1985). A modification of Karmarkar's linear programming algorithm, Manuscript, AT&T Bell Laboratories, Holmdel, New Jersey.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER SOL 86-7	2. GOVT ACCESSION NO. AD A 163648	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) A Note on Nonlinear Approaches to Linear Progrmming		5. TYPE OF REPORT & PERIOD COVERED Technical Report 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Philip E. Gill, Walter Murray, Michael A. Saunders & Margaret H. Wright		8. CONTRACT OR GRANT NUMBER(s) N00014-85-K-0343
9. PERFORMING ORGANIZATION NAME AND ADDRESS Department of Operations Research - SOL Stanford University Stanford, CA 94305-4022		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS NR-047-064
11. CONTROLLING OFFICE NAME AND ADDRESS Office of Naval Research - Dept. of the Navy 800 N. Quincy Street Arlington, VA 22217		12. REPORT DATE April 1986 13. NUMBER OF PAGES 12 pp.
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) UNCLASSIFIED 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

This document has been approved for public release and sale; its distribution is unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

linear programming
Karmarkar's method
barrier methods
sparse least-squares
LU preconditioning

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

See next page.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

SOL 86-7: "A Note on Nonlinear Approaches to Linear Programming,"
Philip E. Gill, Walter Murray, Michael A. Saunders &
Margaret H. Wright

Certain new approaches to linear programming have recently received considerable publicity because of the promise of substantial improvements in efficiency compared to the simplex method. This note briefly discusses several research directions in methods for solving linear programs using nonlinear problem transformations. In particular, we describe application of a barrier transformation to the dual, and the development of sparse least-squares methods based on the LU factorization of the least-squares matrix or its transpose.

END

DTIC

7-86